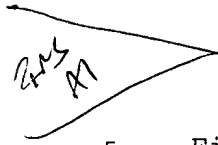


# METHOD FOR INTERACTIVE CONSTRUCTION OF VIRTUAL 3D CIRCUIT MODELS



## 5 Field of the Invention

The present invention relates to a system for computer-aided design (CAD) and particularly for use in a multi-user, real-time collaborative CAD design process.

## Background

The present invention is for use in computer-aided construction or computer-aided design (CAD) and, particularly, mechanical CAD (MCAD). The method is used in simulating assembling and construction of models in virtual environments, whereby designers can try out and visualise different arrangements and interconnections, positioning, etc. and the effects of changes in the positioning or orientation of one component on other components. This enables the designer to make changes during the design process, before a model is actually built.

It has recently become more important that various designers and other participants can work on and view a design simultaneously, during the design process. Most engineering design results from the joint efforts of a highly qualified team of engineers, designers, marketing specialists, etc. The people involved in any particular product design may belong to different companies and may be based in widely differing locations.

To simplify such 'collaborative' design, a number of systems have been developed allowing designers based in different locations to work together over a network in a 'Virtual Design Studio'.

## State of the Art

There are essentially two commonly used types of systems for construction simulation, namely collision  
5 detection and assembly modelling.

The collision detection method starts with tessellated models to achieve real-time capability. The collision detection between one or more moved components and their surrounding serves to identify whether the  
10 component or component group can be built without collision.

A number of disadvantages are, however, associated with collision-based construction processes. In particular, the tessellation results in a loss of  
15 accuracy and a loss in relationship with topological entities.

Furthermore, collision detection can, as the name suggests, only detect when two parts collide, within a given approximation accuracy and cannot determine when  
20 two parts 'fit'.

In assembly modelling processes, at the beginning, and during the construction, 'assembly features' can be added. These assembly features are essentially 'sub' components described in terms of their geometry and also  
25 their nature or behaviour. For example, a 'blind hole' can be defined as an assembly feature which is defined in terms of its geometrical features (length and radius) and also contains information on the direction in which the 'blind hole' is 'open'. This latter information is  
30 often represented by coordinate systems or frames. Such schematic information can be retrieved later for construction. A system related to this process takes into account 'allowable motion'. This provides an efficient means of satisfying 3D constraints by applying  
35 a number of transformation operations to the geometric entity, without invalidating its previously satisfied constraints. The technique of allowable motion is also used for developing an intuitive graphical means of achieving the accurate 3D positioning of a solid model

by automatically constraining its 3D manipulation.

However, using such a system, relationships between components and their topological entity cannot be correctly introduced by user interaction. Furthermore, there is no, or insufficient depiction or visualisation of the degree of freedom which the various components have. Furthermore, a relationship cannot act simultaneously on a group of components.

Anantha et al. ("Assembly modelling by geometric constraint satisfaction", Computer-Aided Design, Vol. 28, No. 9, 1996) describe a way of combining assemblies on the basis of an assembly feature based model. This document teaches the use of analysis of degree of freedom of movement. However, it does not address intuitive interaction and real-time visualisation of the degree of freedom of components by several distributed engineers in a collaborative session.

Various commercial CAD systems are now available for positioning components and constructing circuits or computer models.

Some known systems, e. g. CATIA, IDEAS and Pro/E are constraint based systems.

In constraint based systems, the positioning of the components is defined by constraints or edge conditions defined by the user. Such constraints can be, for example, coincidence, parallelity, etc. The user first defines all of the required constraints and then an algorithm determines whether the various components can be positioned and/or reorientated to satisfy the edge conditions. If not, a message is displayed to the user and the user can then try to re-design to remove any obstructions preventing these conditions being satisfied.

However, a problem with such constraint based systems is that the user must first introduce or define all the required constraints and, only then, does the system see if the components can be positioned in accordance with these constraints. Thus, while the user is actually defining the constraints, he cannot be sure

Other design systems currently available use the non-constraint based technique such as used in CoCreate's SolidDesigner product and IronCAD. SolidDesigner allows components to be positioned or orientated relative to each other by translation and rotation. During the design process, a positioning step can be carried out on a component, but this step in itself does not permanently constrain the position of the component and does not effect its degree of freedom. Thus, with each further transformation, the existing positioning can be changed.

IronCAD goes one step further and offers predefined position characteristics, e. g. surface-on-surface. In the IronCAD system, the positioning step is, again, temporary and can be later undone.

Although many advances have been made in systems for simulating the construction of virtual 3D-CAD models, and the various systems currently in use are very highly developed, they are still not ideally suited for decentralised collaborative real-time construction. They either require generation of construction-relevant data in advance or require knowledge of the construction history, when a change to the design is being made. Alternatively, existing systems may have problems as regards accuracy and are unable to provide to the user the necessary visual information which is required for collaborative design.

The present invention aims to solve these problems by enabling, in the simplest way possible, the assembly or construction of virtual 3D-CAD models which can be simultaneously worked on by a number of participants located in different places.

The system of the present invention provides information to the users which enables them to visualise the possibilities as to positioning and orientation of components in the design.

According to the invention, there is provided a method of changing the relative position and/or orientation of two components of a virtual model displayed on a screen, the method comprising:

- 5       selecting the components whose relative position and/or orientation is to be changed;
- defining the change to be made; and
- calculating and providing on the screen an indication of the remaining degrees of freedom of the
- 10       components after the change has been made.

Preferred embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings.

- Fig. 1 shows possible geometric associations
- 15       between components;

Fig. 2 is a flowchart showing the steps of a method in accordance with the invention;

Fig. 3 shows an example of two objects being positioned in accordance with the invention; and

- 20       Figs. 4 to 11 show different visual feedbacks.

- The invention can preferably be used in the context of one of the existing 3D collaboration design systems such as the OneSpace system of CoCreate. All participants in the design process are connected in a
- 25       network via a 'collaboration server'. The reference model, which is always the current model, is contained in the server and changes made by individual participants to the model are communicated to all of the other participants via the server. Thus, all users see
- 30       the same model at any time and all changes are continuously conveyed to all users. However, the users are able to select any view of the current model.

- To position a component relative to another, the user selects the appropriate topological entities using
- 35       a mouse click, for example, and associates these with each other, e. g. edge of component A to edge of component B. One component acts as a reference element and the other is translated and/or rotated relative to this reference element if its degree of freedom permits,

as discussed below. The role of moved component and reference component can, of course, be changed from operation to operation.

To overcome collisions with other users, when one user is carrying out an operation, the components involved in that operation are 'locked' by the collaboration server, for the duration of the interaction, for all other users.

The components involved in any particular operation are automatically locked, short time, via the collaboration server and indicated to the users by a 'lock feedback'.

After the operation has been completed, the locking on these components is removed and they are available to all participants.

Instead of moving one component at a time, a group of components can be selected and positioned, thus enabling partial assemblies to be moved and positioned.

Topological entities can be associated with each other in the following relationships:

Vertex to vertex

Edge to edge

Axis to axis

Edge to axis and vice versa

Face to face.

Fig. 1 shows some operation possibilities in the form of a tree diagram.

For each combination of degrees of freedom, a vertex, corner, edge or face relationship can be introduced which will reduce, in different ways, the degree of freedom, depending on the geometric constellation of the various associated elements. There are, therefore, 18 (or 21, if the combination 0,0 is included) different possibilities for reducing the degree of freedom.

To start with, before being fixed in any position or orientation, the components all have six degrees of freedom - three degrees of rotational freedom and three degrees of translational freedom. After a positioning

or orientation operation in relation to another component, the number of degrees of freedom may be reduced. Table 1 below shows combinations of degrees of freedom a component can have, after the above operations.

ROTDof	3	3	1	1	1	0	0
TransDof	3	0	2	1	0	1	0

Fig. 2 is a flowchart of the basic process steps when two components or elements are to be associated with each other. First, it is determined whether the desired operation can be fulfilled by transformation of the component within the existing limitations. If so, the component is accordingly transformed. This transformation may involve a reduction in the degrees of freedom. A visual feedback representing the remaining degrees of freedom is calculated, depicted and conveyed to all other users.

Should the users try to carry out operations which cannot be fulfilled within the existing degrees of freedom, they immediately receive a return message that the allocation cannot be permitted.

After making the desired transformation, the new positioning and remaining degrees of freedom are incrementally conveyed to all participants, via the collaboration server, i. e., to reduce the amount of data transmitted, only changes are transmitted.

The visual feedback elements, which show the remaining degrees of freedom after a positioning or orientation operation, serve as 3D operating elements (widgets) which can be clicked on to interactively move a component according to its degree of freedom. In order to obtain precise positioning, movements can be split into discrete steps. The user can determine the discrete steps by inputting values.

The method thus comprises five basic steps:

- checking whether the particular component or group

of components is currently available for positioning. If so, the component or group is locked, via the collaboration server, against use by other users during the operation.

5     -     The association between topological entities is introduced by the user, the appropriate transformation of the component or group of components is automatically calculated by the system.

10    -     To do this, first a check is carried out as to whether the current degree of freedom of that component enables it to be positioned or orientated at all. In a second step, the transformation matrix is calculated which leads to the component which is to be moved taking up the appropriate position and/or orientation.

15    -     A check is carried out as to whether this operation reduces the degree of freedom and, if so, the associated degree of freedom is reduced.

      -     The remaining degree of freedom of the component(s) is calculated and visualised.

20         The locking on that component is then released and the new position and new degrees of freedom are transmitted to all users.

       Fig. 3 shows a particular example of two objects being positioned and orientated relative to each other in accordance with the invention.

25         The two objects A and B have a corner-to-corner or vertex-to-vertex relationship. The designer then wishes to introduce an edge-to-edge relationship. Any other combinations can be carried out in a similar manner.

30         At the start, with the corner-to-corner relationship, the two objects still have three rotational degrees of freedom, but no translational degrees of freedom.

       It is then determined whether or not an edge-to-edge positioning is possible for these two components. It is determined, in this case, that the edge-to-edge association can be precisely carried out by rotation (a translation is not permitted, due to the fact that the components have no translational degrees of freedom), if



the distance between the fixed points, where the two corners are connected, and the edge of the reference component A is equal to the distance between the fixed point and the edge to be moved (of component B).

5        In this case, it is determined that the desired operation is possible and then the transformation matrix for carrying out this operation is determined. In this case, the transformation matrix (here a rotation matrix) is given by the following equation:

10

```
rotAxis=crossProd(vm, vf),
rotAngle=angle(vm, vf) and
fixPoint
```

15

```
with
vm=movedEdge.nearestPoint(fixPoint)-fixPoint
vf=fixEdge.nearestPoint(fixPoint)-fixPoint
```

20

The transformation is then carried out. As this operation results in a reduction in the degrees of freedom of the components, this reduction is then also calculated. The combination of the fixed point and the edge-to-edge association leaves no further movement possibilities for these components, i. e. after the edge-to-edge association, the degree of rotational freedom is

25        also reduced to zero. Thus, in this case, there is no need to calculate and depict a visual feedback showing the remaining degrees of freedom.

30

Should fixedPoint lie on the moved edge and on the fixed edge, then a fixed axis results which includes the fixed point and there then remains a rotational degree of freedom for the combined assembly which can be correspondingly visualised.

35

Figs. 4 to 11 show various visual feedbacks, which show the user the remaining degrees of freedom.

Fig. 4 shows the situation after a surface-to-surface association, showing the visual feedback for two translational and one rotational degrees of freedom.

Fig. 5 shows the situation after an edge-to-edge

association, for example, showing the visual feedback for one translational and one rotational degree of freedom.

Fig. 6 shows the visual feedback for one rotational degree of freedom, e. g. after a corner-to-edge  
5 association.

Fig. 7 shows the visual feedback for one translational degree of freedom, e. g. after an edge-to-surface association.

A further clarifying example is shown in the next  
10 four figures. Here, a 'counterpiece' is inserted into a plate with two holes. In the positioning procedure, first, an axis-to-axis association is carried out.

Fig. 8 shows the situation at the beginning of this process.

15 Fig. 9 shows the situation after the axis-to-axis association, showing a visual feedback element for one translational and one rotational degree of freedom.

In a second interaction step, the two axes are associated by the user and Fig. 10 shows the situation  
20 after this axis-to-axis association. The visual feedback has only changed its position and orientation and shows the user that only one translational degree of freedom remains.

A final surface-to-surface association brings the  
25 components into their final position. Fig. 11 shows the final position of the components after the surface-to-surface association.

Thus, the present invention enables, in the simplest possible way, the assembly or construction of virtual 3D-  
30 CAD models which can be simultaneously worked on by a number of participants at spatially separate locations. By successively considering corners, edges, axes and surfaces of the partial models, these are interactively positioned, orientated and their movement possibilities  
35 are defined. The remaining degrees of freedom are incrementally transmitted to all users and instantaneously visualised. Thus, at every point in time, the distributed users see the same, current model.

Because the users are shown a visual depiction of

the existing degrees of freedom of the part model, at all times, no knowledge of the history of the design process or positioning is needed.

Using the invention, CAD models of different types  
 5 can be assembled or constructed in a simple manner, even by non-CAD system users, in a graphically interactive way and simultaneously by members of a team at different locations.

By successive allocations of corresponding  
 10 topological entities (surfaces, edges, axes and corners), components are 'fixed' with respect to other components, both in position and orientation, in a maximum of three interaction steps.

The method does not make any definition as to the  
 15 application of the model. The required information can be extracted from each 3D model without any additional information such as assembly features having previously been added. Thus, the method of the present invention enables members of a team to quickly, and without  
 20 conflict, position and change the orientation of components relative to each other.

Compared to the known systems described above, the present invention has the following advantages:

- The interaction is targeted, as required by the  
 25 users;
- Relationships between various components are not 'randomly' introduced;
- The effects of associating two components can be immediately visualised by and to all users;
- 30 - The remaining degree of freedom of each component is visually represented to all users;
- Errors and obstructions can be immediately identified, for each operation, and conveyed to the users;
- 35 - There is no need for the user to remember all of the constraints to be introduced and the effects of them;
- No additional information needs to be brought in by the user into the modelling process, as is the case with assembly features;

- 10

1. **Introduction**  
 2. **Background**  
 3. **Methodology**  
 4. **Results**  
 5. **Discussion**  
 6. **Conclusion**  
 7. **References**  
 8. **Appendix**  
 9. **Notes**  
 10. **Tables**  
 11. **Figures**  
 12. **Supplementary Materials**  
 13. **Author Contributions**  
 14. **Funding**  
 15. **Conflict of Interest**  
 16. **Publisher's Note**  
 17. **Copyright**  
 18. **Disclaimer**  
 19. **References**  
 20. **Appendix**  
 21. **Notes**  
 22. **Tables**  
 23. **Figures**  
 24. **Supplementary Materials**  
 25. **Author Contributions**  
 26. **Funding**  
 27. **Conflict of Interest**  
 28. **Publisher's Note**  
 29. **Copyright**  
 30. **Disclaimer**